SAM and the Particle Physics Data Grid

Lauri Loebel-Carpenter, Lee Lueking, Carmenita Moore, Ruth Pordes, Julie Trumbo, Sinisa Veseli, Igor Terekhov, Matthew Vranicar, Stephen White, Victoria White
Computing Division, Fermi National Accelerator Laboratory, Batavia, IL, USA

Abstract
The D0 experiment's data and job management system software, SAM, is an operational prototype of many of the concepts being developed for Grid computing. We explain how the components of SAM map into the Data Grid architecture. We discuss the future use of Grid components to either replace existing components of SAM or to extend its functionality and utility; work being carried out as part of the Particle Physics Data Grid (PPDG) project.

1. Introduction

The D0 data handling system, SAM[1], was built for the "*virtual organization*", D0, consisting of 500 physicists in 72 institutions in 18 countries. Its purpose is to provide a *worldwide* system of shareable computing and storage *resources* that can together be brought to bear to solve the *common problem* of extracting physics results from about a Petabyte of measured and simulated data (c.2003). The goal of the system is to provide a large degree of *transparency* to the user who makes *requests* for datasets (*collections*) of relevant data and *submits jobs* that execute monte-carlo simulation, reconstruction or analysis programs on available computing resources. Transparency in storage and delivery of data is currently in a more advanced state than transparency in the submission of jobs. Programs executed, in the context of SAM, *transform data* by consuming data file(s) and producing resultant data file(s) of a different data content or 'data tier'. Data files are read-only and are never modified, or versioned.

These data handling and job control services, typical of a data grid, are provided by a collection of servers using CORBA communication. The software components are D0-specific prototypical implementations of some of those identified in Data Grid Architecture documents [3] [4][5]. Some of these components will be replaced by 'standard' data grid components emanating from the various grid research projects, including PPDG, [2] [5] [6]. Others will be modified to conform to Grid protocols and APIs. Additional functional components and services will be integrated into the SAM system. This work forms the D0/SAM component of the Particle Physics Data Grid project.

2. The Fabric

The *fabric* on which SAM currently operates consists of *compute systems* and *storage systems* at Fermilab, France/Lyon-IN2P3, UK/Lancaster, Netherlands/Nikhef, Czech Republic/Prague, US/UTA, US/Columbia, US/MSU, UK/Imperial College. Many other sites are expected to provide additional compute and storage resources when the experiment moves from commissioning to physics data taking. Storage systems consist of *disk storage elements* at all locations and robotically controlled tape libraries at Fermilab, Lyon and Nikhef - with Lancaster about to be added to this list. All storage elements support the basic functions of storing or retrieving a file. Some support parallel transfer protocols, currently via bbftp [8]. The underlying storage management systems for *tape storage elements* are different at Fermilab, Lyon and Nikhef. Currently only the Fermilab tape storage management system, Enstore [7], provides the ability to assign priorities and file placement instructions to file requests and provides reports about placement of data on tape, queue wait time, transfer time and other information that can be used for resource management.

Catalogs of *metadata*, locations(*replica catalog*), *data transformations* (processing history), SAM configuration and policy, as well as databases of detector calibration, detector configuration, and other parameters, are implemented as Oracle relational databases. Other information required to characterize and reproduce *data products,* refers to versioned code and parameter files residing in a cvs *code repository*.

Whereas Babar and the LHC experiments subdivide their fabric into centers at specific geographic locations (eg. Tier0, Tier 1, Tier A), the D0 fabric is organized into physical groupings of compute, storage and network resources that are operated and managed together for a common purpose. These are termed

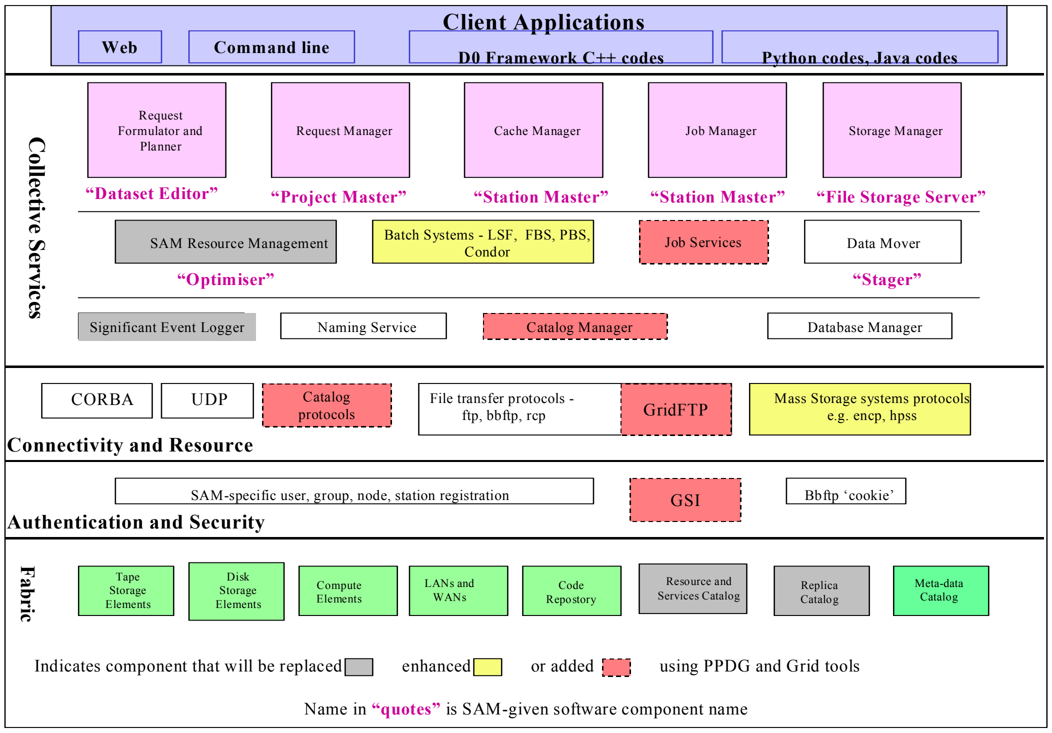
"Stations". At Fermilab there are currently six production SAM Stations named: datalogger, d0farm, central-analysis, linux-build-1, linux-analysis-cluster-1 and clued0 (a cluster of linux desktop machines). Tape storage elements are designated as directly accessible from certain stations. Their access from other stations must be defined via a route through one or more SAM stations which provide *caching* and *forwarding* services. There is an accessibility matrix for each compute element and each storage element managed or accessed by a Station.[11]

Disk storage elements may be either Station or externally managed. Station-managed elements together create logical disk caches that are administered for a particular physics *group*. While all files residing in the Station's disk storage elements are in fact accessible by any group permitted to use the Station resources, it is expected that the files for one physics group will be largely disjoint from those of another and, therefore, caching and "*pinning*"of files proceed according to the separate *quotas* and *policies* of each physics group.

Although Stations are intended to own resource partitions they do not necessarily have exclusive control over them. For example, on the 176-node SGI system at Fermilab, at times, we run both a 'central-analysis' station and a 'central-archive' station, sharing compute elements, but with distinct and disjoint disk storage elements. This also allows us to run "development" or "test" grids that share compute elements and tape storage systems but have separate files, catalogs and station-managed disk.

3. SAM Architecture and Components

In the diagram below the software components of SAM are illustrated and categorized in dependent layers, according to recent documents that attempt to characterize the architecture of a Grid or Data Grid system [3],[4],[5]. We identify (**in bold**) various components of the SAM system that will be either be extended, completely replaced, or added to the system as a result of work carried out by PPDG and other grid projects and the integration of that work into the SAM system, as part of the PPDG work plan.



4. Connectivity and Resource Services

The current system uses rather primitive user identification and authentication mechanisms. Unix user names, SAM physics groups, nodes, domains and stations are registered and maintained centrally in the SAM configuration catalog. Valid combinations of these must be provided to obtain services. The bbftp [8] file transfer protocol implements its own form of authentication. Station servers at one station provide service on behalf of their local users and are 'trusted' by other Station servers or Database Servers. **Globus core Security Infrastructure services is a planned PPDG enhancement of the system.** [12]

Service registration and discovery is implemented using a CORBA naming service, with namespace by station name. APIs to services in SAM are all defined using CORBA Interface Definition Language and have multiple language bindings (C++, Python, Java) and, in many cases, a shell interface. This includes services that provide access to information in various catalogs. UDP is used for significant event logging services and for certain *Request Manager* control messages. Rcp, Kerberized rcp, bbftp and encp[7] provide file transfer protocols. Work is currently underway to provide a more scalable and robust CORBA naming service, possibly by replicating the naming service in multiple locations and adding persistency. **Use of GridFTP and other standard protocols to access storage elements is a planned PPDG modification to the system**. **Integration with grid monitoring tools and approaches is a PPDG area of research. Registration of resources and services using a standardized Grid registration or enquiry protocol, in addition to the current mechanism, is a PPDG enhancement to the system.**

5. Software Components for Collective Services

Database Servers provide access to the *Replica Catalog*, *Metadata Catalog*, SAM Resource and configuration catalog and *Transformation catalog*. Currently multiple database servers provide services. They may run an any machine with SQLnet. Each is capable of providing all catalog services, but the workload is partitioned administratively. All catalogs currently have their persistent format as tables in a central Oracle database; a matter that is hidden from their clients. Replication of some catalogs in two or more locations worldwide is a planned enhancement to the system.
Database servers will need to be implemented that adapt SAM-specific APIs and catalog protocols onto Grid catalog APIs using PPDG-supported Grid protocols so that information may be published and retrieved in the wider Physics Data Grid that spans several virtual organizations.

A central **Logging server** receives significant events. This will be refined to receive only summary level information, with more detailed monitoring and significant event information held at each site. **Work in the context of PPDG will examine how to use a Grid Monitoring Architecture and tools.**

Each disk storage element has a **Data Mover** ("Stager") associated with it that provides services to transfer or erase a file by using the appropriate protocol for the source and destination storage elements involved.

Resource manager services are provided by an "Optimization" service. File transfer actions are prioritized and authorized prior to being executed. **The current primitive functionality of re-ordering and grouping file requests, primarily to optimize access to tapes, will need to be greatly extended, redesigned and re-implemented to better deal with co-location of data with computing elements and fair-shares and policy-driven use of all computing, storage and network resource. This is a major component of the SAM/PPDG work, to be carried out in collaboration with the Condor team [15].**

A **Request Formulator and Planner** "Dataset Definition Editor" provides the user with a natural and mathematical language formulation of queries based on metadata. Immediate feedback on the size of the resultant *file set* is given, but feedback on the likely time to gain access to that dataset is not yet available.

Each Station **Cache Manager** provides caching services, including the ability to "pin" files, and implements the designated caching and "pinning" policies for each physics group.

A Station **Job Manager** provides services to execute a user application, script, or series of jobs, potentially as parallel processes, either interactively or by use of a local batch system. Currently supported batch systems are LSF[13] and FBS[14], with adapters for PBS and Condor[15] in preparation and test. The station Cache Manager and Job Manager are implemented as a single "Station Master" server.
Job submission and synchronization between job execution and data delivery is currently part of SAM. Jobs are put on hold in batch system queues until data files are available to the job. At present jobs submitted at one station may only be run using the batch system(s) available at that Station.

The specification of user jobs, including their characteristics and input datasets, is a major component of the PPDG work. The intention is to provide Grid job services components that replace the SAM job services components . This will support job submission (including composite and parallel jobs) to suitable SAM Station(s) and eventually any available Grid computing resource.

Request Managers arrange for pre-staging of file replicas and handle all bookkeeping of file consumption, including errors, retries and restarts. This "Project Master" server, under the control of the "Station Master"; is executed for each dataset that is to be delivered and consumed by a user job. Each Request manager, working together with its Station Cache Manager and Data Movers, together implement a robust *file replication service*, for files of a requested dataset, that are not in accessible Station storage elements, for the particular compute elements involved. This file replication service handles inter-Station transfers of data and intra-Station transfers between storage elements. Each station currently has a 'preferred location' for its files, with non-preferred locations, if used, chosen randomly.
PPDG Resource Management components are needed, along with Grid job services components, to provide optimized and cost effective replication and to assure co-location of jobs and data.

Storage Manager services are provided by a Station's "File Storage Server" that responds directly to user commands to store files in tape and disk storage elements. It also services requests from other "File Storage Servers". All file storage requests must be accompanied by metadata that identifies not only the nature of the file and its data contents, but also the parent files and the processing history by which this file was obtained. For example a typical client storage command would be "sam store –descrip=my-meta.py" [--dest=]". Metadata parameters for various data products are currently implemented using Python classes and are extensible. Final storage destinations may be automatically determined based on the file metadata, or may be explicitly specified. Access control is by user, group and station and is not yet as stringent as necessary. In an attempt to encourage use of the system we have probably left it too open for users to store files in the system, and on tape.

6. Client Applications

SAM APIs have multiple language bindings and so Client Applications may be user commands, web applications, C++ reconstruction or analysis code or Python or Java code. In keeping with the goal of transparent data access, the D0 C++ application framework[16] hides the data handling system from the user. It provides access to files in a dataset (in random order) via the same I/O mechanisms used for access to single named files[17]. A designator SAMInput: or SAMOutput: achieves this and metadata for output files is created. Richer metadata is required in the future for all data products. The metadata catalog itself now supports an expandable and rich set of physics-related information, but the production jobs and monte-carlo jobs right now provide only limited metadata; surely less than required for full support of *virtual data*. Work is underway to make a ROOT client application with transparent access to SAM files. With expanded metadata will come capabilities for a more physics-friendly dataset definition language and tools. Users submit processing or analysis jobs via commands such as "sam submit –script=mystuff –dataset=mydataset –group=mygroup –jobpars=…..".

7. Robustness and Scalability – Experiences so far

Almost all of the Monte Carlo data produced so far has been created on Farm nodes outside of Fermilab and the resulting reconstructed or root-tuple files (more than 20TB) have been stored using the File Storage Server mechanism described above - from stations at Nikhef, Lancaster, Prague and Lyon. The system is in constant use for Online system data logging (OSF1), farm production processing at Fermilab on a 90 node Linux Farm system, and for multiple purposes of monitoring, analysis and program debugging on the central analysis machine (SGI-IRIX) and several clusters of Linux workstations. The remote sites are ready and able to use their Stations to retrieve either Raw or processed physics data from Fermilab. More details about the operation of the whole system are given in other papers presented at this conference [9] [18]. Typically more than 30 datasets are simultaneously being served up on "central-analysis", with file replication only as needed, within the 2.4 TB of disk cache.

There are currently single points of failure in the system and these must be eliminated. Recently the implementation of the CORBA naming service that we chose has proved unreliable and troublesome. The

Online System's data logging station uses an optional mode of operation where physical movement of files may proceed, even if the catalogs are not available; cataloging requests are queued. The central Oracle database server, although clearly a single point of failure and in the end non-scalable, has proved to be robust and highly available. Plans to replicate certain catalogs must nevertheless continue. Users have been given considerable freedom to execute 'requests' that may result in complex and lengthy database queries. We need to do more work to prevent wild queries and also to provide additional database server connections to the database to handle peak demands – that can happen when, for example, all students in a tutorial define a gigantic dataset at the same moment. The file replication is not completely watertight against all forms of failure and certain bad behavior of certain file transfer protocols that may return success, even in the face of failure. Encouragingly, we have observed robust behavior following network failures while storing files from Nikhef to Fermilab, where the process halted and then seamlessly continued after a configurable retry period. Tuning the various retrial and other parameters is currently more of an art, than a science and would be greatly aided by a more sophisticated Grid Monitoring environment, as well as a better user interface for administrators to adjust the various tuning knobs.

One of the most difficult tasks has proved to be the evolution of the system while maintaining backwards compatibility and without enforcing that all Stations upgrade to the newest version of software simultaneously. That is a challenge, that we have met only some of the time.

8. Conclusions

The system is operational and working quite well and can be viewed, including browsing most of its catalogs, installation guides, users guides and administrators guides, at http://d0db.fnal.gov/sam. Integration of standard grid components, via PPDG, will greatly enhance the functionality of the system, while providing a real-life, in-use, vertically integrated Grid system as a testbed for such components.

REFERENCES

1. The SAM home page http://d0db.fnal.gov/sam
2. The Particle Physics Data Grid (PPDG) home page: http://www.ppdg.net/
3. I. Foster, C. Kesselman, S. Tuecke, *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*. To be published in Intl. J. Supercomputer Applications, 2001.
4. *Data Grid Reference Architecture*. at http://www.ppdg.net/docs/documents_and_information.htm
5. *DataGrid Architecture* at http://www.eu-datagrid.org/
6. The GriPhyN project home page: http://www.griphyn.org/
7. The Enstore home page. http://www-isd.fnal.gov/enstore
8. The Babar parallel file transfer protocol bbftp home page: http://doc.in2p3.fr/bbftp/
9. L. Carpenter *et.al.*, SAM Overview and Operation at the D0 Experiment. Submitted to CHEP2001, September, 2001.
10. I. Terekhov *et. al.*, *SAM for D0 – a fully distributed data access system*. Talk presented at VII International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2000), Oct, 2000, Proceedings.
11. I. Terekhov *et. al.*, *Distributed Data Access and Resource Management in the D0 SAM System*. The Tenth IEEE International Symposium on High Performance Distributed Computing San Francisco, California, August 7-9, 2001, in Proceedings.
12. The Globus Project home page: http://www.globus.org/
13. LSF workload management from Platform Computing. http://www.platform.com/
14. I. Mandrichenko *et. al.*, *Farms Batch System and Fermi Inter-Process Communication toolkit*, CHEP 2000 proceedings, FERMILAB-CONF-00-083
15. The Condor project home page: http://www.cs.wisc.edu/condor
16. J. Kowalkowski, *The D0 Framework*. Talk presented at CHEP2000, February 2000, Padova, Italy
17. The D0 object model. http://wwwd0.fnal.gov/d0dist/dist/releases/test/d0om/doc/html/d0om_user_guide/
18. V.White *et.al. The D0 Data Handling System. Submitted to CHEP2001*, September, 2001